

IN THE CLAIMS

Please amend claims 1, 7 and 13, as follows.

1           1.       **(Currently Amended)** A computer-based method for determining the optimum  
2 join sequence for processing a query having a plurality of tables from a relational database stored  
3 in an electronic storage device having a database management system, the method comprising  
4 the steps of:

5           (a) a first pass using simulation for determining an optimum join sequence for joining the  
6 plurality of tables from the query; and

7           (b) a second pass for using the optimum join sequence for creating a lowest cost access  
8 path plan for processing the query.

1           2.       **(Original)** The method according to claim 1, wherein the first pass performing  
2 successive steps until creation of a simulated composite table having all tables from the query,  
3 wherein each said step:

4           creating a set of miniplans for simulating all possible joins of a predetermined subset of  
5 the query tables; and

6           using a cost model calculations for estimating and saving the least expensive join from  
7 said set of joins, thereby determining the optimum join sequence.

1           3.       **(Original)** The method according to claim 2, wherein the first pass for each said  
2 miniplan storing a used table index, join method, and sorting data, and for each said least  
3 expensive join storing names of joined tables, join cost and possible row orderings.

1           4.       **(Original)** The method according to claim 3, wherein the first pass only storing  
2 non-redundant miniplan data, and saving partial results of the cost model calculations for future  
3 reuse.

1           5.       **(Original)** The method according to claim 1, wherein the second pass performing  
2 successive steps until creation of a simulated composite table having all tables from the query,  
3 wherein each said step being performed in the optimum join sequence.

1           6.       **(Original)** The method according to claim 1, wherein the query being a SQL  
2 query.

1           7.       **(Currently Amended)** A computer-based processor system for determining the  
2 optimum join sequence for processing a query having a plurality of tables from a relational  
3 database stored in an electronic storage device having a database management system, the  
4 system comprising:

5               means for performing a first pass using simulation for determining an optimum join  
6 sequence for joining the plurality of tables from the query; and

7               means for performing a second pass for using the optimum join sequence for creating a  
8 lowest cost access path plan for processing the query.

1           8.     **(Original)** The system according to claim 7, wherein the first pass means  
2 performing successive steps until creation of a simulated composite table having all tables from  
3 the query, wherein each said step:  
4           creating a set of miniplans for simulating all possible joins of a predetermined subset of  
5 the query tables; and  
6           using a cost model calculations for estimating and saving the least expensive join from  
7 said set of joins, thereby determining the optimum join sequence.

1           9.     **(Original)** The system according to claim 8, wherein the first pass means for each  
2 said miniplan storing a used table index, join method, and sorting data, and for each said least  
3 expensive join storing names of joined tables, join cost and possible row orderings.

1           10.    **(Original)** The system according to claim 9, wherein the first pass means only  
2 storing non-redundant miniplan data, and saving partial results of the cost model calculations for  
3 future reuse.

1           11.    **(Original)** The system according to claim 7, wherein the second pass means  
2 performing successive steps until creation of a simulated composite table having all tables from  
3 the query, wherein each said step being performed in the optimum join sequence.

1           12.    **(Original)** The system according to claim 7, wherein the query being a SQL  
2 query.

1           13.   **(Currently Amended)** A computer usable medium tangibly embodying a  
2   program of instructions executable by the computer to perform a computer-based method for  
3   determining the optimum join sequence for processing a query having a plurality of tables from a  
4   relational database stored in an electronic storage device having a database management system,  
5   the method comprising the steps of:

6           (a) a first pass using simulation for determining an optimum join sequence for joining the  
7   plurality of tables from the query; and

8           (b) a second pass for using the optimum join sequence for creating a lowest cost access  
9   path plan for processing the query.

1           14.   **(Original)** The method according to claim 13, wherein the first pass performing  
2   successive steps until creation of a simulated composite table having all tables from the query,  
3   wherein each said step:

4           creating a set of miniplans for simulating all possible joins of a predetermined subset of  
5   the query tables; and

6           using a cost model calculations for estimating and saving the least expensive join from  
7   said set of joins, thereby determining the optimum join sequence.

1           15.   **(Original)** The method according to claim 14, wherein the first pass for each said  
2   miniplan storing a used table index, join method, and sorting data, and for each said least  
3   expensive join storing names of joined tables, join cost and possible row orderings.

1           16.   **(Original)** The method according to claim 15, wherein the first pass only storing  
2 non-redundant miniplan data, and saving partial results of the cost model calculations for future  
3 reuse.

1           17.   **(Original)** The method according to claim 13, wherein the second pass  
2 performing successive steps until creation of a simulated composite table having all tables from  
3 the query, wherein each said step being performed in the optimum join sequence.

1           18.   **(Original)** The method according to claim 13, wherein the query being a SQL  
2 query.